

Índices no Oracle Database

Tudo o que você precisa saber sobre eles
(em 50 minutos)

Fábio Prado

www.fabioprado.net

Apresentação

Quem sou eu?

- Trabalho com TI há 16 anos:
 - 2000/2001: Analista de Suporte;
 - 2001/2007: Analista Sistemas/Desenvolvedor;
 - Desde 2007: **DBA**;
 - Desde 2009: **Instrutor**
- **Instrutor** dos seguintes cursos em Bancos de Dados:
 - SQL e PL/SQL;
 - Administração de Bancos de Dados Oracle;
 - **Tuning** (de SQL, PL/SQL e Banco de Dados).
- Bacharel em Ciências da Computação, com MBA e Pós-graduação em Gestão de Projetos;

Quem sou eu?

- Autor do blog [FABIOPRADO.NET](http://www.fabioprado.net)
(<http://www.fabioprado.net>);
- Articulista da revista SQL Magazine e diversos sites e blogs de TI;
- Certificações e títulos:
 - Microsoft: MCP, MCSD, MCAD, MCSD.NET, **MCDBA**, MCTS, MCT e MCPD;
 - Oracle: **Oracle ACE**; [OCE SQL Tuning](#); [OCE Database Performance Tuning](#), OCA PL/SQL, OCP Database 10G e 11G.

Introdução

- A apresentação irá abordar as dúvidas mais comuns sobre índices no Oracle Database através de **perguntas, respostas e exemplos.**

**1- O que é mais rápido:
FTS ou IS?**

1- O que é mais rápido: FTS ou IS?

FULL TABLE SCAN

Leitura da tabela inteira, do início (1º bloco) ao fim (até a HWM)

INDEX SCAN

Também conhecido como index lookup, consiste na pesquisa através de valores chaves em estruturas de dados adicionais que podem conter (na maior parte dos índices utilizados) os dados da coluna + ROWID

1- O que é mais rápido: FTS ou IS?



1- O que é mais rápido: FTS ou IS?

RESPOSTA = **DEPENDE**

- Depende de uma série de fatores, tais como:
 - Tamanho da tabela:
 - Em tabelas pequenas normalmente um FTS é mais rápido.
 - Seletividade da consulta:
 - Para retornar poucos dados (até 4% segundo a Oracle, em colunas com alta cardinalidade) normalmente um IS é mais rápido.

1- O que é mais rápido: FTS ou IS?

Planilha Query Builder

```
1 EXPLAIN PLAN FOR
2 SELECT CD_CLIENTE, NM_CLIENTE, DT_NASCIMENTO
3 FROM ECOMMERCE.CLIENTE
4 WHERE NM_EMAIL = 'sagko@com.br';
5 SELECT * FROM TABLE(dbms_xplan.DISPLAY);
```

Saída do Script x

Tarefa concluída em 0,031 segundos

Plan FOR bem-sucedido.

Todas as Linhas Extraídas: 64073 em 1,47 segundos

PLAN_TABLE_OUTPUT

Plan hash value: 1138695813

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	36	274 (1)	00:00:01
* 1	TABLE ACCESS FULL	CLIENTE	1	36	274 (1)	00:00:01

Predicate Information (identified by operation id):

PLAN_TABLE_OUTPUT

1 - filter("NM_EMAIL"='sagko@com.br')

FTS com custo de 274 executado em 1,47s

1- O que é mais rápido: FTS ou IS?

Planilha Query Builder

```
1 -- criando um indice btree na coluna NM_EMAIL da tabela ECOMMERCE.CLIENTE
2 CREATE INDEX ECOMMERCE.IX_CLIENTE_NMEMAIL ON ECOMMERCE.CLIENTE (NM_EMAIL);
3
4 -- gerando o PE depois de criar o indice p/ comparar com o 1o. PE
5 EXPLAIN PLAN FOR
6 SELECT CD_CLIENTE, NM_CLIENTE, DT_NASCIMENTO
7 FROM ECOMMERCE.CLIENTE
8 WHERE NM_EMAIL = 'sagko@com.br';
9 SELECT * FROM TABLE(dbms_xplan.DISPLAY);
10
```

Resultado da Consulta x

Todas as Linhas Extraídas: 14 em 0,047 segundos

PLAN_TABLE_OUTPUT

1 Plan hash value: 2496080830

2 Todas as Linhas Extraídas: 1 em 0,004 segundos

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	36	3 (0)	00:00:01
1	TABLE ACCESS BY INDEX ROWID BATCHED	CLIENTE	1	36	3 (0)	00:00:01
2	INDEX RANGE SCAN	IX_CLIENTE_NMEMAIL	1		1 (0)	00:00:01

Predicate Information (identified by operation id):

2 - access ("NM_EMAIL"='sagko@com.br')

IS com custo de 3 executado em 0,004s
Alta seletividade + alta cardinalidade + btree = índice bom
Tempo de execução caiu de 1,47s p/ 0,004s (36.750% + rápido)

1- O que é mais rápido: FTS ou IS?

Planilha Query Builder

```
1 create table ecommerce.cliente2 as select * from ecommerce.cliente;
2 UPDATE ECOMMERCE.CLIENTE2 SET NM_EMAIL = 'sagko@com.br';
3 COMMIT;
4 CREATE INDEX ECOMMERCE.IX_CLIENTE_NMEMAIL2 ON ECOMMERCE.CLIENTE2 (NM_EMAIL);
5 EXEC DBMS_STATS.GATHER_TABLE_STATS('ECOMMERCE','CLIENTE2');
6 ALTER SESSION SET OPTIMIZER_MODE = FIRST_ROWS;
7
8 -- gerando o PE depois de criar o indice p/ comparar com o 1o. PE
9 EXPLAIN PLAN FOR
10 SELECT CD_CLIENTE, NM_CLIENTE, DT_NASCIMENTO, NM_EMAIL
11 FROM ECOMMERCE.CLIENTE2
12 WHERE NM_EMAIL = 'sagko@com.br';
13 SELECT * FROM TABLE(dbms_xplan.DISPLAY);
```

Todas as linhas da tabela possuirão o mesmo e-mail.

Forçando o uso do índice "ruim"

Todas as Linhas Extraídas: 64073 em 1,766 segundos

Saída do Script x Resultado da Consulta x

SQL | Todas as Linhas Extraídas: 14 em 0,016 segundos

PLAN_TABLE_OUTPUT

1 Plan hash value: 2875713482

2

3

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		64067	2252K	1194 (1)	00:00:01
1	TABLE ACCESS BY INDEX ROWID BATCHED	CLIENTE2	64067	2252K	1194 (1)	00:00:01
2	INDEX RANGE SCAN	IX_CLIENTE_NMEMAIL2	64067		215 (0)	00:00:01

11 Predicate Information (identified by operation id):

12

13

14 2 - access("NM_EMAIL"='sagko@com.br')

IS com custo de 1194 executado em 1,766s
Baixa seletividade + baixa cardinalidade + btree = **índice ruim**
Tempo de execução aumentou de 1,47s p/ 1,766s (**20% pior**)

2- Quando devemos criar índices?

2- Quando devemos criar índices?

- Índices **otimizam consultas**, mas **degradam a performance de instruções DML** (insert, update, delete e merge), portanto, crie-os criteriosamente, seguindo algumas das dicas abaixo:
 1. Para otimizar **consultas frequentes**;
 2. Nas colunas da cláusula **WHERE**, ordenação, agrupamentos e ligações;
 3. Em geral, quando a **consulta retorna poucas linhas** da(s) tabela(s);
 4. Em FKs **para otimizar joins** e DMLs na tabela pai;
 5. Quando o SQL não tiver uma **cláusula de negação**:
 - ✓ Exemplos: **NOT IN**, **<>**, **!=**, **NOT EXISTS**.
 6. Quando o filtro não comparar **valores nulos** (NULL), pois eles não são indexados nos índices mais comuns (B-tree).

3- Quais os tipos de índices existentes e quando devo utilizar cada um deles?

3- Quais os tipos de índices existentes e quando devo utilizar cada um deles?

- Os índices são subdivididos em 4 grupos:

1. B-tree:

- Alta cardinalidade;
- Índices default, também conhecidos como “normais”, são muito utilizados em ambientes OLTP.

```
-- criando um indice btree na coluna NM_EMAIL da tabela ECOMMERCE.CLIENTE
CREATE INDEX ECOMMERCE.IX_CLIENTE_NMEMAIL ON ECOMMERCE.CLIENTE (NM_EMAIL);

EXPLAIN PLAN FOR
  SELECT CD_CLIENTE, NM_CLIENTE, DT_NASCIMENTO
  FROM   ECOMMERCE.CLIENTE
  WHERE  NM_EMAIL = 'sagko@com.br';
SELECT * FROM TABLE(dbms_xplan.DISPLAY);
```

3- Quais os tipos de índices existentes e quando devo utilizar cada um deles?

2. Bitmap:

- Foi introduzido no Oracle 7.3 e só pode ser criado na versão **Enterprise Edition**;
- Mais indicado para **baixa cardinalidade** e ambientes onde **não há alta concorrência de gravações**;
- Podem ser utilizados para comparações com NULL e são muito utilizados em ambientes OLAP.

```
1  -- criando indice bitmap na coluna ID_STATUS da tabela ECOMMERCE.PEDIDO
2  CREATE BITMAP INDEX ECOMMERCE.IX_PEDIDO_IDSTATUS ON ECOMMERCE.PEDIDO(ID_STATUS);
3  /
4
5  -- gerando PE depois de criar o indice bitmap
6  EXPLAIN PLAN FOR
7  SELECT CD_PEDIDO, DT_PEDIDO
8  FROM   ECOMMERCE.PEDIDO
9  WHERE  ID_STATUS = 5;
10 SELECT * FROM TABLE(dbms_xplan.DISPLAY);
11 /
12
```

3- Quais os tipos de índices existentes e quando devo utilizar cada um deles?

3. Function Based:

- Foi introduzido no Oracle 8i e pode ser btree ou bitmap;
- Deve ser criado quando for necessário usar função(es) na(s) coluna(s) do(s) filtro(s).

```
-- criando indice FB bitmap (mais apropriado p/ esta consulta)
CREATE BITMAP INDEX ECOMMERCE.IX_PEDIDO_DTPEDIDO ON ECOMMERCE.PEDIDO (TO_CHAR(DT_PEDIDO, 'YYYY'));

-- gerando o PE depois de criar o indice FB bitmap
EXPLAIN PLAN FOR
  SELECT    COUNT(1)
  FROM      ECOMMERCE.PEDIDO
  WHERE     TO_CHAR(DT_PEDIDO, 'YYYY') = '2009';
SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);
```

3- Quais os tipos de índices existentes e quando devo utilizar cada um deles?

4. Application Domain:

- Índices especializados para otimizar consultas mais complexas;
- Exemplos: *Oracle Text* e *Oracle Spatial*.

```
1 -- Execute o comando abaixo para criar o índice IX_NO_CLIENTE_OT:
2 create index ECOMMERCE.IX_NO_CLIENTE_OT on ECOMMERCE.CLIENTE (NM_CLIENTE) indextype is ctxsys.context;
3
4 -- Para testar o acesso ao índice, execute:
5 EXPLAIN PLAN FOR
6   SELECT *
7   FROM   ECOMMERCE.CLIENTE
8   WHERE  CONTAINS(NM_CLIENTE, '%ul%') > 0;
9 SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);
```

**4- Criei um índice, porque
ela não está sendo
utilizado?**

4- Criei um índice, porque ele não está sendo utilizado?

a) A tabela é pequena:

```
1 create table tab_nf (cd_nf varchar2(100), dt_nf date, vl_nf number);
2 insert into tab_nf (cd_nf, dt_nf, vl_nf) values ('1', sysdate, 1000);
3 insert into tab_nf (cd_nf, dt_nf, vl_nf) values ('1b', sysdate, 1000);
4 commit;
5 create index ix_tabnf_cdnf on tab_nf(cd_nf);
6
7 explain plan for
8   select *
9   from   tab nf
10  where  cd_nf = '1';
11 select * from table(dbms_xplan.display);
12
```

A tabela TAB_NF contém somente 2 linhas

Resultado da Consulta 1 x

SQL | Todas as Linhas Extraídas: 17 em 0,014 segundos

PLAN_TABLE_OUTPUT

1 Plan hash value: 705547230

2

3 -----

4	Id	Operation	Name	Rows	Bytes	Cost (\$CPU)	Time
5							
6	0	SELECT STATEMENT		1	74	2 (0)	00:00:01
7	* 1	TABLE ACCESS FULL	TAB_NF	1	74	2 (0)	00:00:01
8							

4- Criei um índice, porque ele não está sendo utilizado?

- b) O(s) objeto(s) possui(em) estatística(s) desatualizada(s):

```
7 begin
8   for i in 1..999999
9     loop
10      insert into tab_nf (cd_nf, dt_nf, vl_nf) values (to_char(i), sysdate, to_char(i));
11    end loop;
12    commit;
13  end;
14
15  explain plan for
16    select *
17    from   tab_nf
18    where  cd_nf = '1';
19  select * from table(dbms_xplan.display);
20
```

Saída do Script x | Resultado da Consulta x | Resultado da Consulta 1 x | Resultado da Consulta 2 x

SQL | Todas as Linhas Extraídas: 13 em 0,021 segundos

PLAN_TABLE_OUTPUT

1 Plan hash value: 705547230

2

3 -----

4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
5							
6	0	SELECT STATEMENT		1	14	2 (0)	00:00:01
7	* 1	TABLE ACCESS FULL	TAB_NF	1	14	2 (0)	00:00:01
8							

4- Criei um índice, porque ele não está sendo utilizado?

- b) O(s) objeto(s) possui(em) estatística(s) desatualizada(s):

```
15 explain plan for
16   select *
17   from   tab_nf
18   where  cd_nf = '1';
19 select * from table(dbms_xplan.display);
20
21 exec dbms_stats.GATHER_table_stats(USER,'TAB_NF');
22
23 explain plan for
24   select *
25   from   tab_nf
26   where  cd_nf = '1';
27 select * from table(dbms_xplan.display);
28
```

Saída do Script x | Resultado da Consulta x | Resultado da Consulta 1 x | Resultado da Consulta 2 x | Resultado da

SQL | Todas as Linhas Extraídas: 14 em 0,022 segundos

PLAN_TABLE_OUTPUT

1 Plan hash value: 1082614265

2

3

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	20	4 (0)	00:00:01
1	TABLE ACCESS BY INDEX ROWID BATCHED	TAB_NF	1	20	4 (0)	00:00:01
* 2	INDEX RANGE SCAN	IX_TABNF_CDNF	1		3 (0)	00:00:01

11 Predicate Information (identified by operation id):

12

Após coletar estatísticas o índice passou a ser utilizado

4- Criei um índice, porque ele não está sendo utilizado?

- c) Existe alguma **conversão implícita/explicita** no SQL:

```
1 explain plan for
2   select *
3   from   tab_nf
4   where  cd_nf = '1';
5 select * from table(dbms_xplan.display);
6
```

Saída do Script x Resultado da Consulta 5 x

SQL | Todas as Linhas Extraídas: 14 em 0,167 segundos

PLAN_TABLE_OUTPUT

1 Plan hash value: 1082614265

2

3 -----

4 Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
5						
6 0	SELECT STATEMENT		1	20	4 (0)	00:00:01
7 1	TABLE ACCESS BY INDEX ROWID BATCHED	TAB_NF	1	20	4 (0)	00:00:01
8 *	2 INDEX RANGE SCAN	IX_TABNF_CDNF	1		3 (0)	00:00:01
9						
10						
11	Predicate Information (identified by operation id):					
12	-----					
13						
14	2 -	access("CD_NF"='1')				

4- Criei um índice, porque ele não está sendo utilizado?

- c) Existe alguma **conversão implícita/explicita** no SQL:

```
1 explain plan for
2   select *
3   from   tab_nf
4   where  cd_nf = 1;
5 select * from table(dbms_xplan.display);
6
```

Saída do Script x | Resultado da Consulta 5 x | Resultado da Consulta 6 x

SQL | Todas as Linhas Extraídas: 13 em 0,031 segundos

PLAN_TABLE_OUTPUT

1 Plan hash value: 705547230

2

3 -----

4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
5							
6	0	SELECT STATEMENT		1	20	940 (1)	00:00:01
7	* 1	TABLE ACCESS FULL	TAB_NF	1	20	940 (1)	00:00:01
8							
9							

10 Predicate Information (identified by operation id):

11 -----

12

13 1 - filter(TO_NUMBER("CD_NF")=1) ➡ conversão implícita

4- Criei um índice, porque ele não está sendo utilizado?

- d) O índice possui um alto **fator de clusterização**:

```
SQL> create table organized
2 as
3 select x.*
4   from (select * from stage
5         order by object_name) x
6 /
Table created.
```

Tabela ordenada alfabeticamente

```
SQL> create table disorganized
2 as
3 select x.*
4   from (select * from stage
5         order by dbms_random.random) x
6 /
Table created.
```

Tabela sem ordenação

```
SQL> create index organized_idx on organized(object_name);
Index created.

SQL> create index disorganized_idx on disorganized(object_name);
Index created.

SQL> begin
2  dbms_stats.gather_table_stats
3  ( user, 'ORGANIZED',
4    estimate_percent => 100,
5    method_opt=>'for all indexed columns size 254'
6  );
7  dbms_stats.gather_table_stats
8  ( user, 'DISORGANIZED',
9    estimate_percent => 100,
10   method_opt=>'for all indexed columns size 254'
11 );
12 end;
13 /
PL/SQL procedure successfully completed.
```

4- Criei um índice, porque ele não está sendo utilizado?

- d) O índice possui um alto fator de clusterização:

```
SQL> select table_name, blocks, num_rows from user_tables
2  where table_name like '%ORGANIZED' order by 1;
```

TABLE_NAME	BLOCKS	NUM_ROWS
DISORGANIZED	1064	72839
ORGANIZED	1064	72839


```
SQL> select table_name, index_name, clustering_factor
2  from user_indexes
3  where table_name like '%ORGANIZED' order by 1;
```

TABLE_NAME	INDEX_NAME	CLUSTERING_FACTOR
DISORGANIZED	DISORGANIZED_IDX	72760
ORGANIZED	ORGANIZED_IDX	1038

Índices possuem mesma qtde. de blocos e linhas

Somente ORGANIZED_IDX possui baixo fator de clusterização

```
SQL> select * from organized where object_name like 'A%';
```

Id	Operation	Name	Rows	Bytes	Cost
0	SELECT STATEMENT		1825	174K	39
1	TABLE ACCESS BY INDEX ROWID	ORGANIZED	1825	174K	39
* 2	INDEX RANGE SCAN	ORGANIZED_IDX	1825		12

```
SQL> select * from disorganized where object_name like 'A%';
```

Id	Operation	Name	Rows	Bytes	Cost
0	SELECT STATEMENT		1825	174K	291
* 1	TABLE ACCESS FULL	DISORGANIZED	1825	174K	291

Fonte: On Clustering Factor and Validating Keys, Tom Kyte

Link: <http://www.oracle.com/technetwork/issue-archive/2012/12-sep/o52asktom-1735913.html>, acessado em 01/04/2016 17:40h

www.fabioprado.net

4- Criei um índice, porque ele não está sendo utilizado?

- e) Índice ineficiente. Ex.: Btree criado em coluna com baixa cardinalidade:

```
1 update tab_nf set vl_nf = 1000;
2 commit;
3
4 create index ix_tabnf_vlnf on tab_nf(vl_nf);
5
6 exec dbms_stats.GATHER_table_stats(USER,'TAB_NF');
7
8 explain plan for
9   select *
10  from   tab nf
11  where  vl_nf = 1000;
12 select * from table(dbms_xplan.display);
```

Coluna “vl_nf” possui baixa cardinalidade

```
1 Plan hash value: 705547230
2
3 -----
4 | Id | Operation          | Name | Rows | Bytes | Cost (%CPU)| Time     |
5 -----
6 | 0 | SELECT STATEMENT   |      | 999K | 17M   | 938  (1) | 00:00:01 |
7 |* 1 | TABLE ACCESS FULL | TAB_NF | 999K | 17M   | 938  (1) | 00:00:01 |
8 -----
9
10 Predicate Information (identified by operation id):
11 -----
12
13 1 - filter("VL_NF"=1000)
```

5- Posso forçar o uso de um índice?

5- Posso forçar o uso de um índice?

- Sim, você pode forçar o uso de índices, mas tenha **CUIDADO: você poderá estar forçando o uso de um índice ineficiente;**
- Para usá-los, ao invés de utilizar um hint INDEX, utilize alguma das opções abaixo:
 - `ALTER SESSION SET OPTIMIZER_MODE=FIRST_ROWS;`
 - Hint `FIRST_ROWS`.
- Na configuração padrão do parâm `OPTIMIZER_MODE`, alguns índices “bons” podem ser ignorados.

Obs.: Se isso não for suficiente altere o valor de outros parâmetros que influenciam o comportamento do otimizador. Ex.: `OPTIMIZER_INDEX_COST_ADJ`.

5- Posso forçar o uso de um índice?

```
8 explain plan for
9   select *
10  from   tab_nf
11  where  vl_nf = 1000;
12 select * from table(dbms_xplan.display);
13
```

PLAN_TABLE_OUTPUT

1 Plan hash value: 705547230

2

3

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		999K	17M	938 (1)	00:00:01
* 1	TABLE ACCESS FULL	TAB_NF	999K	17M	938 (1)	00:00:01

```
14 explain plan for
15   select /*+ FIRST ROWS */ *
16  from   tab_nf
17  where  vl_nf = 1000;
18 select * from table(dbms_xplan.display);
19
```

PLAN_TABLE_OUTPUT

1 Plan hash value: 759985989

2

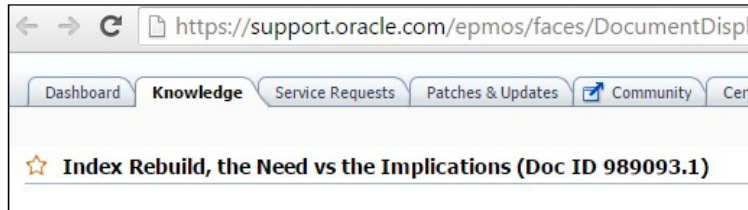
3

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		999K	17M	5395 (1)	00:00:01
1	TABLE ACCESS BY INDEX ROWID BATCHED	TAB_NF	999K	17M	5395 (1)	00:00:01
* 2	INDEX RANGE SCAN	IX_TABNF_VLNF	999K	1958	(1)	00:00:01

Índice ineficiente

6- Quando devemos reconstruir índices?

6- Quando devemos reconstruir índices?



6- Quando devemos reconstruir índices?



Referências

- **Expert Indexing in Oracle Database 11G -**
Bill Padfield, Darl Kuhn, Ignatius Fernandez, Sam Alapati -
Apress;
- **Expert Oracle Database Architecture -** Thomas Kyte - Apress
- Treinamento [SQL Tuning](#) [FABIOPRADO.NET](#)

FIM

contato@fabioprado.net